"How's the neural network project going?"

ASSIGNMENT 5

ITRI 626

ENRICO DREYER
31210783

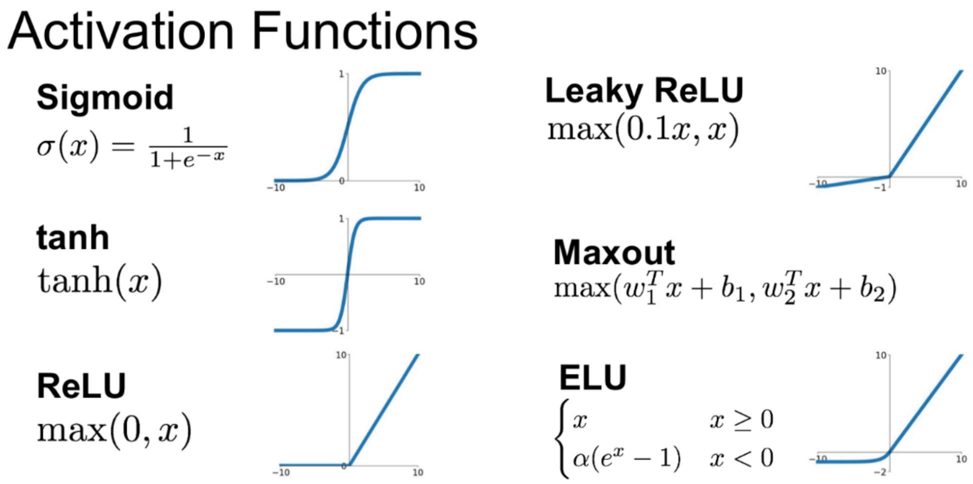
Table of Contents

List of figures

Introduction

For this assignment we were asked to write a report on neural network activations functions in general. According to (Sibi et al., 2013) activation functions in neural networks are used to transform a signal input to a signal output, when in turn is used as input to the next layer of the neural network.

Why activation functions are needed will be discussed as well as the types of activation functions, along with a visual representation of each. After doing the practical assignment a discussion will be made on which activation function gave the smallest error. Below is a representation of some of the activation functions.



Why neural network need activation functions?

If no activation function is used in neural networks the output signal would be a simple linear function (Sharma et al., 2017). A linear equation is easy and simple to solve but the complexity of that neural network is limited, and they do not have the ability to recognize and learn complex data.

For this reason, there is a need for activation functions as well as the use of artificial neural network such as Deep Learning that can make use of complicated, non-linear datasets and high dimensional (Kukreja et al., 2016). The neural network model can also have hidden layers as well as complex architecture that is used for extracting knowledge (Sibi et al., 2013).

Types of activation functions

Having the ability to limit amplitude of an output of a neuron and giving it a limited range, is called squashing functions. Using a squashing function squashes the amplitude of the neural network output signal to a specific finite value. Different types are as follows:

- Hyperbolic tangent
- Linear
- ReLu
- Sigmoid

Hyperbolic tangent function

Hyperbolic Tangent function is also known as Tanh function. The Tanh function is similar to that of the sigmoid function with the difference being that the symmetric being around the origin. The results of the different signs of outputs from a previous layer is fed to the next input. The Hyperbolic tangent function can be defined as: f (x) = 2sigmoid(2x)-1.

The tanh function is differentiable and continuous, the values are between -1 and 1. Compared to the sigmoid function is that the gradient of tanh is steeper, as shown in Figure 1.

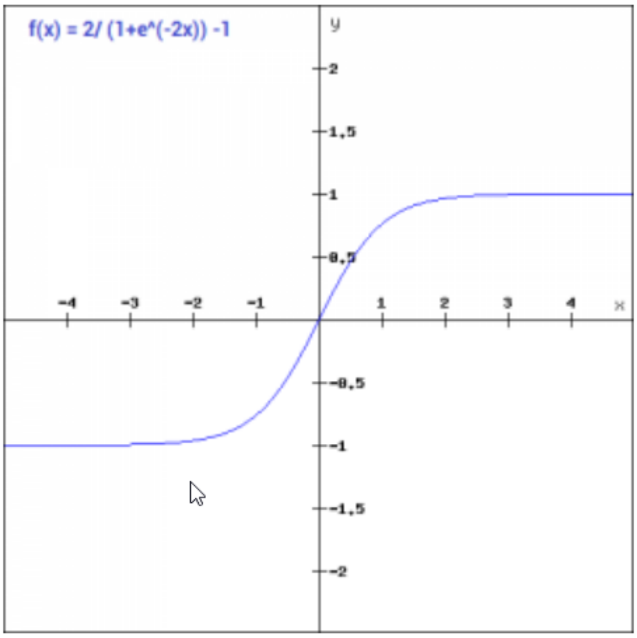


Figure 1: Hyperbolic tangent function

Linear function

Using a linear activation function will give you an output that is directly proportional to the input given. The drawback of using this function is that the binary step function has no gradient, meaning that there is no component for x in a binary step function.

The linear activation function can be defined as F(x) = ax. An example of a linear activation graph is shown in Figure 2.

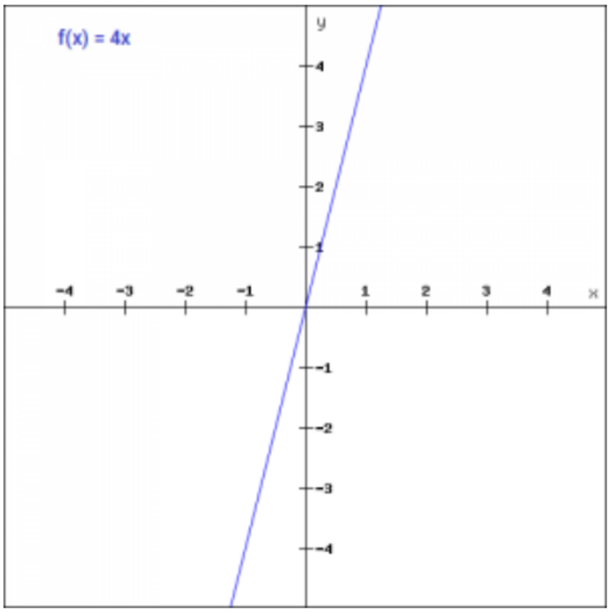


Figure 2: Linear activation (Sharma et al., 2017)

ReLU function

Rectified liner unit function is a function defined as non-linear that is widely used in neural networks. Using ReLU gives you the upper hand in terms of neurons being able to be activated at different times (Sharma et al., 2017). This means that a neuron will only be deactivated only when an output is zero.

ReLU can be mathematically defined as f(x) = max(0,x). An example of a ReLU activation graph is shown in Figure 3.

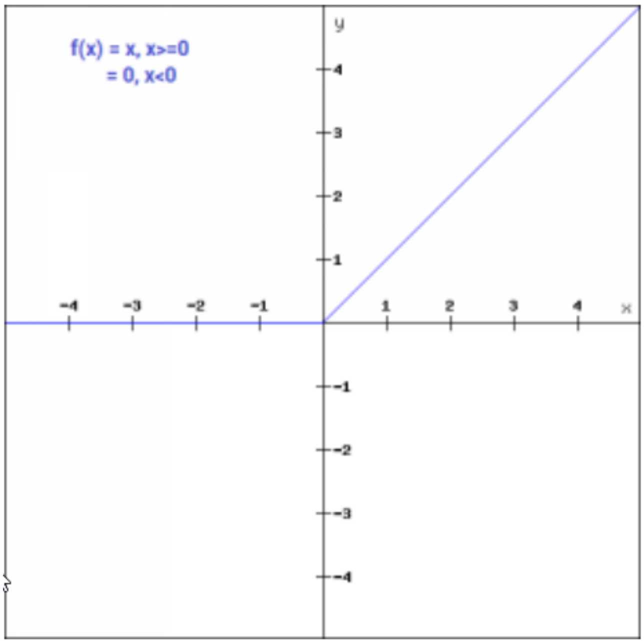


Figure 3: ReLU function (Sharma et al., 2017)

Sigmoid function

The sigmoid function is the most used function because it is a non-linear function. This function transforms values into a value between 1 and 0 and can be mathematically defined as f(x) = 1/e-x. Sigmoid function is a smooth S-shaped as shown in Figure 3 and is continuously differentiable.

The derivative of the sigmoid function is f'(x) = 1-sigmoid(x). The sigmoid function is also not symmetric, this means that for all the output values of the signs will be the same (Sibi et al., 2013).

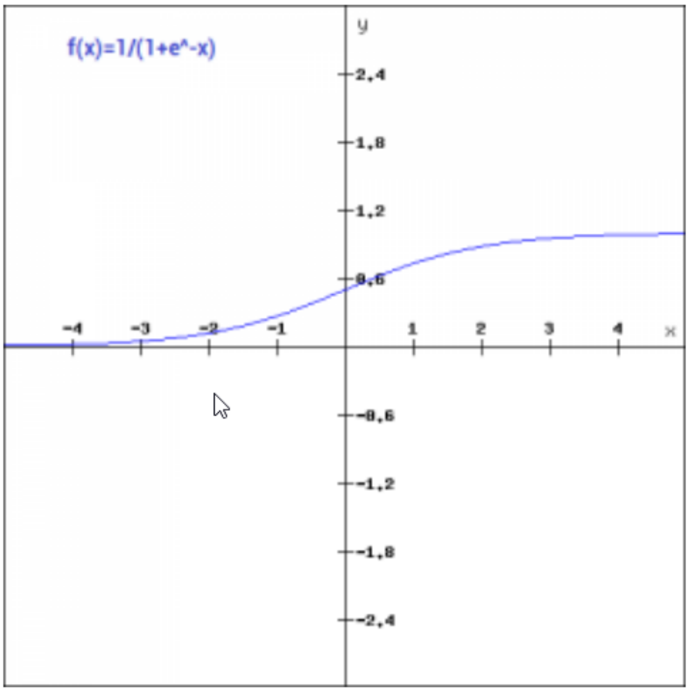
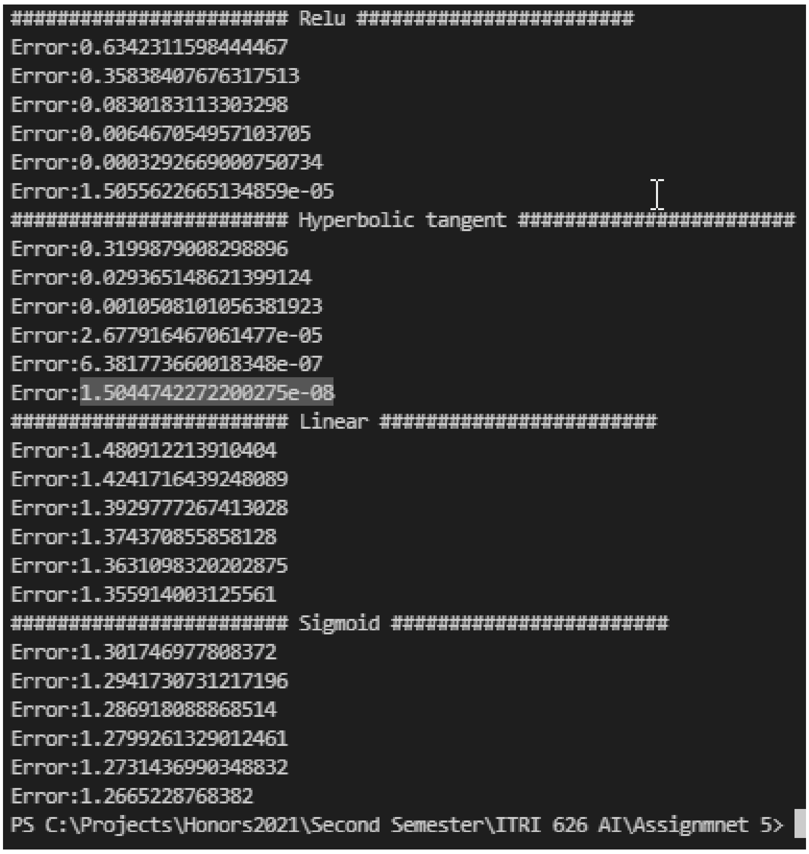


Figure 4: Sigmoid function (Sharma et al., 2017)

What has the lowest error based on the practical assignment?

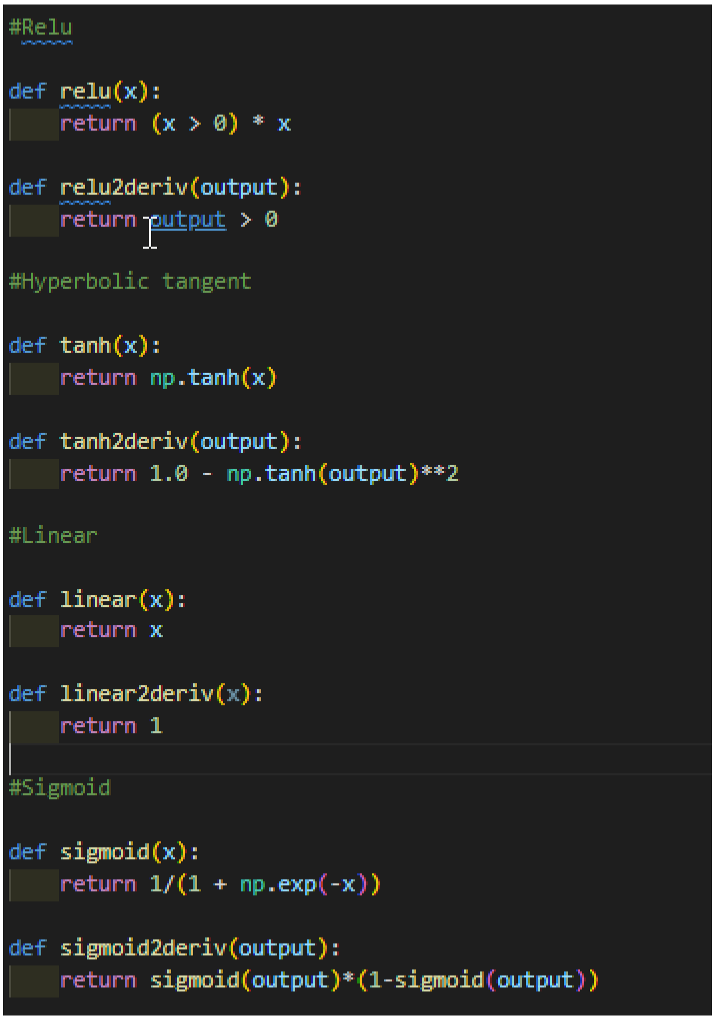
According to the results from the practical assignment the activation function with the lowest error was Hyperbolic tangent as it gave an error of "1.5044742272200275e-08" as shown below in Figure 5.



```
######################## Relu ########################
Error:0.6342311598444467
Error:0.358384076763175l3
Error:0.0830183113303298
Error:0.006467054957103705
Error:0.0003292669000750734
Error:1.5055622665134859e-05
######################## Hyperbolic tangent ########################
Error:0.3199879008298896
Error:0.029365148621399124
Error:0.0010508101056381923
Error:2.677916467061477e-05
Error:6.381773660018348e-07
Error:1.5044742272200275e-08
######################## Linear ########################
Error:1.480912213910404
Error:1.4241716439248089
Error:1.3929777267413028
Error:1.374370855858128
Error:1.3631098320202875
Error:1.355914003125561
######################## Sigmoid ########################
Error:1.301746977808372
Error:1.2941730731217196
Error:1.286918088868514
Error:1.2799261329012461
Error:1.2731436990348832
Error:1.2665228768382
PS C:\Projects\Honors2021\Second Semester\ITRI 626 AI\Assignmnet 5>
```

Figure 5: Code snippet

Below in Figure 6 is the two methods used for calculating the four activation functions for layer 1 as well as their derivative (Myonn, 2018).



```python
#Relu

def relu(x):
    return (x > 0) * x

def relu2deriv(output):
    return output > 0

#Hyperbolic tangent

def tanh(x):
    return np.tanh(x)

def tanh2deriv(output):
    return 1.0 - np.tanh(output)**2

#Linear

def linear(x):
    return x

def linear2deriv(x):
    return 1

#Sigmoid

def sigmoid(x):
    return 1/(1 + np.exp(-x))

def sigmoid2deriv(output):
    return sigmoid(output)*(1-sigmoid(output))
```

Figure 6: Functions snippet

Conclusion

For this assignment we were asked to write a report on neural network activations functions in general. Why activation functions are needed were discussed as well as the types of activation functions, along with a visual representation of each. The activation function with the lowest error was the hyperbolic tangent function.

References

Kukreja, H., Bharath, N., Siddesh, C., & Kuldeep, S. (2016). An introduction to artificial neural network. *Int J Adv Res Innov Ideas Educ*, *1*, 27-30.

Myonn, C. (2018). Writing Activation Functions From (Mostly) Scratch in Python. https://cup-of-char.com/writing-activation-functions-from-mostly-scratch-in-python/

Sharma, S., Sharma, S., & Athaiya, A. (2017). Activation functions in neural networks. *towards data science*, *6*(12), 310-316.

Sibi, P., Jones, S. A., & Siddarth, P. (2013). Analysis of different activation functions using back propagation neural networks. *Journal of theoretical and applied information technology*, *47*(3), 1264-1268.